

NASA/TM—2001-210818



Web-Based Distributed Simulation of Aeronautical Propulsion System

Desheng Zhang
AYT Corporation, Brook Park, Ohio

Gregory J. Follen
Glenn Research Center, Cleveland, Ohio

William R. Pavlik
AYT Corporation, Brook Park, Ohio

Chan M. Kim
Glenn Research Center, Cleveland, Ohio

Xianyou Liu
AYT Corporation, Brook Park, Ohio

Tammy M. Blaser and Isaac Lopez
Glenn Research Center, Cleveland, Ohio

Prepared for the
10th International Symposium on High Performance Distributed Computing
sponsored by the Institute of Electrical and Electronics Engineers
San Francisco, California, August 7–9, 2001

National Aeronautics and
Space Administration

Glenn Research Center

April 2001

Acknowledgments

The authors would like to thank the Glenn AvSP team and IPG-CORBA team for numerous design discussions and the support of the NASA AvSP program. We also express appreciation to Frances J. Lawas-Grodek for setting up the working environment and to Allen P. Holtz for helping set up Globus environments at NASA Glenn.

This report contains preliminary findings, subject to revision as analysis proceeds.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100

Available electronically at <http://gltrs.grc.nasa.gov/GLTRS>

Web-Based Distributed Simulation of Aeronautical Propulsion System

Desheng Zhang,^a Gregory J. Follen,^b William R. Pavlik,^a Chan M. Kim,^b
Xianyou Liu,^a Tammy M. Blaser,^b and Isaac Lopez^b

^aAYT Corporation, Brook Park, Ohio 44142

^bNational Aeronautics and Space Administration, Glenn Research Center, Cleveland, Ohio 44135

Abstract

An application was developed to allow users to run and view the Numerical Propulsion System Simulation (NPSS) engine simulations from web browsers. Simulations were performed on multiple INFORMATION POWER GRID (IPG) test beds. The Common Object Request Broker Architecture (CORBA) was used for brokering data exchange among machines and IPG/Globus for job scheduling and remote process invocation. Web server scripting was performed by JavaServer Pages (JSP). This application has proven to be an effective and efficient way to couple heterogeneous distributed components.

Introduction

The goal of National Airspace System (NAS)'s Aviation Safety Program (AvSP) is to “develop and demonstrate technologies that contribute to a reduction in aviation accident and fatality rates by a factor of 5 by year 2007 and by a factor of 10 by year 2022”[1]. To achieve this, various simulation systems are used to provide risk assessment and performance parameters of aircraft arriving at and departing from airports. In particular, NASA Glenn Research Center's Numerical Propulsion System Simulation (NPSS) [2-4] will process flight data and provide engine parameters. In this process, the engine simulation module has to handle a huge amount of flight data and send the simulation results back to the user across geographic boundaries. A typical use case may have a user who initiates processing of a particular data set gathered for flights at a specific airport, submits data for engine simulation, and gets the results in graphical/tabular format. In this case, the flight data, user interface, and simulation module may be hosted by different machines and even by different platforms. In order to achieve this scenario, one needs to consider a computational architecture that can incorporate a ubiquitous user interface, a heterogeneous computing environment and a high performance computing power.

In this work, an n-tier application was developed to address these issues. The user submits simulation requests through a web interface. CORBA/Java technologies are used to build middle-tier components. In the back-end, NPSS is used to process flight data and provide engine parameters. In order to handle the intensive computational load, NASA's IPG [5] test beds and Globus [6,7] were used. In all, our application achieves a web-based, multi-platform, high-performance computing environment.

The following sections describe the components of the system in detail, demonstrate the simulation process with various screen snapshots and conclude with future work.

Architecture

Fig. 1 shows the overall architecture of engine simulation. It consists of several machines: (1) machines with browser; (2) web server machine hosting JSP, database, flat files, a CORBA client and a CORBA server; (3) coupling machine hosting a CORBA server and a CORBA client; (4) IPG test beds hosting NPSS engine application and a CORBA client.

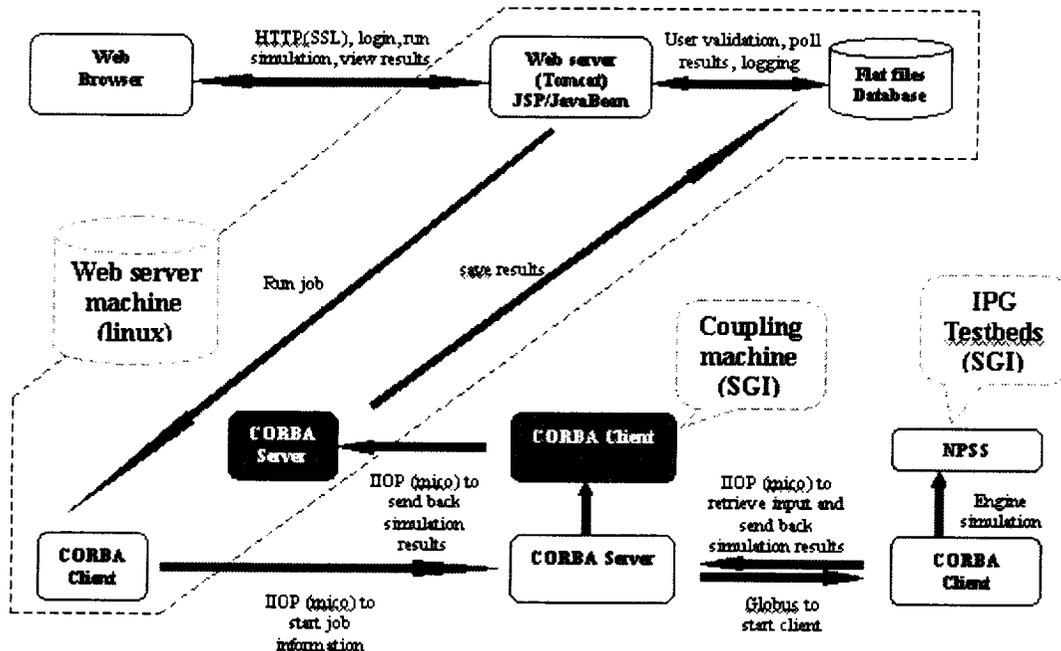


Figure 1: Engine simulation architecture

System Elements

Web browser, web server and server scripting:

The advantage of developing a web-based application is that a user can access it from anywhere. To run our application properly, the browser needs to have applets enabled.

Tomcat, a subproject of the open-source Jakarta project, was used as JSP/Servlet engine server [8]. Version 3.2 of Tomcat supports JSP 1.1 and Servlet 1.2 specifications. Because Java is the default and only supported scripting language, it offers several advantages such as easy programming, easy database access through JDBC, and code portability.

Tomcat is very easy to maintain. Two shell scripts were provided to start and stop the server. There are two main XML files for configurations: server.xml for the whole server and web.xml for each application. Besides running standalone, Tomcat can also run as an Apache module. When properly configured, Apache delegates .jsp requests to Tomcat for handling. This allows many features of Apache to be leveraged, such as SSL.

We used JSP files to deliver dynamic web content [9]. JSP files may be slow in their first use but later invocation is much faster. JSP does much of its real work in JavaBean, such as database access and data processing. This allows short JSP files and code reusing.

Database and flat files:

Cloudscape 3.0, written in pure Java, is a reliable, high performance DBMS [10] and can be installed on many platforms. It provides three frameworks: embedded, RmiJdbc and Cloudconnector as well as the corresponding JDBC drivers. We used the RmiJdbc framework to support multiple connections.

To provide better performance, both flight input data and engine simulation results were stored in flat files while simulation configuration data were kept in the database. One reason for this design is that flight input data and simulation results are seldom used for query. On the other hand, simulation configuration data are frequently used to look up information. Another reason is to avoid data loss. The average number of flight input records is approximately 250. There will be about 250 0-Dimensional output records and additional 1000 1-Dimensional records if zooming option is selected. This corresponds to 1250 SQL insert statements that must be done in one transaction. If a user ran simulations for two or more flights in a very close time period, data loss for one or more flights was sometimes observed.

A unique job id is assigned to each simulation session that may include one or more flights from the same SMA date and one engine model. To easily tell the content of the flat file, the file name contains the combination of flight id, flight type (arrival or departure) and SMA date. While flight input files are shared, simulation result files are stored under a job id subdirectory which allows the same flight input to be used to run multiple simulations with the same or different configuration(s).

CORBA:

CORBA is the ideal tool to develop distributed applications. All CORBA products support IIOP protocol. The open source MICO was used as our C++ implementation. MICO works on many platforms and supports the CORBA 2.3 specification, especially the Portable Object Adaptor (POA) [11]. Our MICO applications run on both LINUX and SGI-IRIX (IPG test beds) platforms. Firewall configuration is not required.

Among all the machines used, the coupling machine is the hub. First, the web server machine forwards the user's job request through a CORBA client to the CORBA server on the coupling machine. The CORBA server then splits the job into sub-jobs, if necessary, by referring to the start-up configuration information and using Globus to invoke remote CORBA clients on IPG test beds to actually run the sub-jobs. Once the simulation is finished, the results are sent back to the CORBA server. The coupling machine stores both flight input and simulation results. To make web presentation more efficient, the web server machine has its own local copies of both flight input and simulation results. It has a CORBA server running all the time to receive simulation results from the coupling machine. The CORBA server on the coupling machine has an embedded CORBA client to do this task.

When the CORBA server on the coupling machine starts up, it runs a start-up script that contains the IPG test beds configuration information including grid label, grid DNS name, root directory of NPSS application and workload. The information is used to allocate workload and locate remote CORBA client.

Because there may be multiple CORBA clients to request services from the CORBA server at the same time, a servant manager of type servant locator was used to provide parallel processing. Upon start-up, the server creates 20 inactive objects. At runtime, job id is used as a key to use one of the inactive objects. The servant manager then binds the object to a newly created servant to serve the request. After serving the request, the servant is deleted and the object is inactivated again and reused to serve other requests.

Globus and IPG test beds:

The Globus project is developing fundamental technologies needed to build computational grids [5-7]. It allows grid users to run jobs on remote grid machines. To achieve this, grid users need do three things first: (1) generate a private key and request a Globus certificate; (2) install

the private key file and certificate file at home directories on all grids they want to use; (3) have their contact string added into Globus mapfiles on all grids they want to use. Globus always uses a scheduler such as LSF and PBS to run jobs. In fact, on all grid machines, it is mandatory to use a scheduler for long-running jobs.

Globus allows us to harness the computational power of grid machines. For our application, each flight simulation may take 1-2 minutes. If users want to run 2000 flights (average number of flights per day at a large airport), it will take two days to run on a single grid machine. If users want to add zooming to their simulation, more time will be needed. By splitting the workload onto several grid machines, it is practical to have these simulations done overnight.

Besides invoking remote CORBA clients, Globus is also used to deliver updated IORs. Each time the CORBA server on the coupling machine restarts, CORBA clients need to be aware of the newly generated IORs. Globus is used to send these IORS to IPG test beds as well as the web browser machine, thus avoiding manual copying of IORs or using a naming service. The maintenance of the CORBA server is simply applying for the Globus proxy and starting the CORBA server. As mentioned above, there is a CORBA server running on the web server machine. When it starts up, it sends its IORs to a web location. The coupling machine is able to get the IORs by running a Perl script.

Engine model:

NPSS v1.0 was used to run the generic turbofan engine simulation. The model is made of elements such as flight conditions, inlet, fan, compressor, burner, turbines, shafts and nozzle. The input of the NPSS executable are ground temperature and radar track records (x coordinate, y coordinate, range, azimuth, velocity, mach, temperature, timestamp). Radar data is available within a fifty-mile radius of the airport. The calculated simulation output parameters include fan shaft speed, high speed shaft rpm, compressor inlet & outlet temperatures and pressures, turbine inlet & outlet temperatures and pressures, fuel rate, burner efficiency, thrust and CO emission. In addition, 1-Dimensional zooming data including temperature and pressure of both compressor and turbine were generated by using a linear algorithm.

The CORBA clients on the IPG test beds invoke an engine simulation through a shell script that sends several parameters including job id and flight id to the NPSS executable. For each radar track record, one 0-Dimensional output record is generated. Flights in a job and radar-track records of a flight are processed sequentially. Each radar track record is treated independently.

Demonstration

A demonstration of the application is available only to users with an assigned user id and a valid password. Visitors can click the visitor button to go to a visitor page to view a sample of the simulation data including tabular flight input, tabular simulation output, X-Y plots of the 0D engine data (applets) and 1D compressor pressure zooming animation (applet).

Users who want to run an engine simulation or view history jobs have to login to the system through the login page. Passwords are encrypted using a one-way hash function and allowed to be changed by the user. After login, the user can either view history jobs or select flights to run engine simulation.

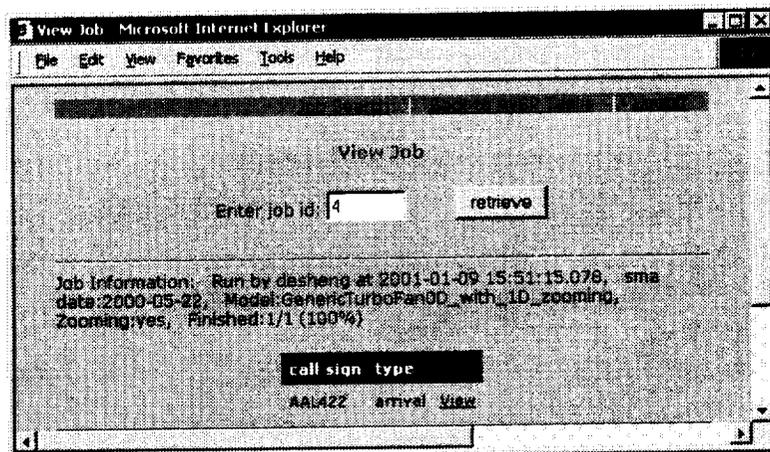


Figure 2: Job information page.

time received	altitude (ft)	mach #	inlet temp (F)	lispd rpm	Fan rpm	HPT TempIn (F)	HPT TempOut (F)	HPT PrsIn (psi)	HPT PrsOut (psi)	HPT TempIn (F)	HPT TempOut (F)	HPT PrsIn (psi)	HPT PrsOut (psi)	hrnr eff	fuel rate (lb)	thrust (lb)	CO Emission
969013342	22,600	0.831	-1.0	7,790.0	5,162.0	295.0	789.0	28.0	145.0	2,085.0	1,014.0	131.0	37.0	0.999	5.04	29,788.0	2,345.0
969013348	22,500	0.826	-1.0	7,790.0	5,162.0	295.0	789.0	28.0	145.0	2,094.0	1,018.0	131.0	37.0	0.999	5.00	29,794.0	2,345.0
969013351	22,400	0.821	-1.3	7,781.0	5,162.0	295.0	788.0	27.0	145.0	2,279.0	1,021.0	130.0	37.0	0.999	5.01	29,517.0	2,345.0
969013355	22,000	0.814	-1.7	7,770.0	5,162.0	291.0	782.0	27.0	144.0	2,273.0	1,019.0	130.0	36.0	0.999	4.99	29,287.0	2,345.0
969013360	22,100	0.819	0.0	7,762.0	5,162.0	289.0	780.0	27.0	144.0	2,267.0	1,019.0	130.0	36.0	0.999	4.93	29,321.0	2,345.0
969013365	22,000	0.823	0.3	7,767.0	5,162.0	289.0	778.0	27.0	143.0	2,254.0	1,007.0	129.0	34.0	0.999	4.91	29,335.0	2,345.0
969013369	21,800	0.824	1.0	7,759.0	5,162.0	291.0	779.0	27.0	144.0	2,265.0	1,004.0	129.0	34.0	0.999	4.99	29,742.0	2,345.0
969013374	21,800	0.824	2.1	7,767.0	5,162.0	290.0	781.0	27.0	144.0	2,270.0	1,019.0	129.0	37.0	0.999	5.00	29,692.0	2,345.0
969013378	21,200	0.821	3.2	7,774.0	5,162.0	292.0	784.0	28.0	147.0	2,274.0	1,017.0	129.0	37.0	0.999	5.06	29,402.0	2,345.0
969013383	20,900	0.813	4.2	7,777.0	5,162.0	282.0	784.0	28.0	146.0	2,274.0	1,018.0	129.0	38.0	0.999	5.10	29,525.0	2,345.0

Figure 3: Tabular 0D engine data.

If the user knows the job id, this can be entered in a textbox to retrieve the job information including user id, run time, SMA date, model id, status and the list of flights in the job. Otherwise, a job search can be performed by specifying a query of combination of user id, SMA date and run date or simply request showing all jobs. The user selects the job id link to go to the job information page (Fig. 2). From the job information page, select the link for the flight to navigate to the flight detail page where the tabular input, tabular output (Fig. 3), tabular zooming, X-Y plots of 0D engine data (applets, Fig. 4, 5), 1D temperature and pressure zooming animation of both compressor and turbine (applets, Fig. 6), and download input and output data can be viewed.

The user starts the engine simulation by selecting a SMA date (Fig. 7). Then, a list of flights with type of either arrival or departure on that date is displayed in a list box (Fig. 8). The user can select the flights one by one, or select all arrival flights, or select all departure flights, or

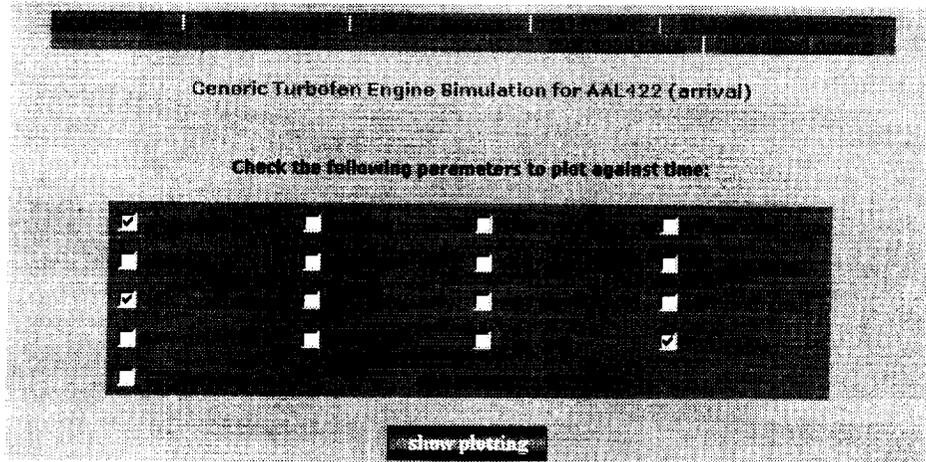


Figure 4: Engine parameters available to plot against time step.

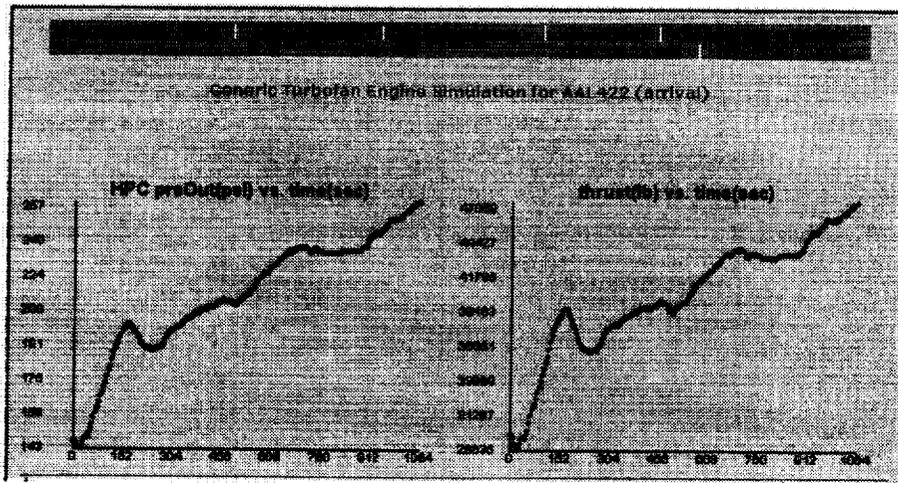


Figure 5: Applet showing X-Y plot of engine parameters.

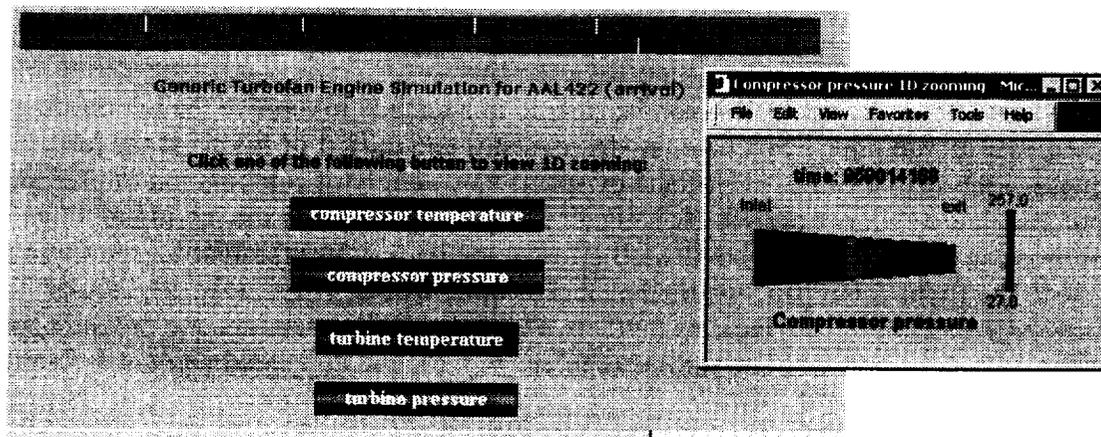


Figure 6: Applet showing 1D zooming.

select all flights. There are two engine models available for selection: one is a 0-Dimensional model without 1-Dimensional zooming and the other is a 0-Dimensional model with 1-Dimensional zooming (using a linear algorithm). The selected flights, flight types, model id and SMA date specify a job, which is labeled with a unique job id. The Engine simulation of the job is triggered by clicking the simulation button. An applet keeps updating to show the progress of the job (Fig. 9). From there, the user clicks the job id link to go to the job information page and the remaining operations, which were described above.

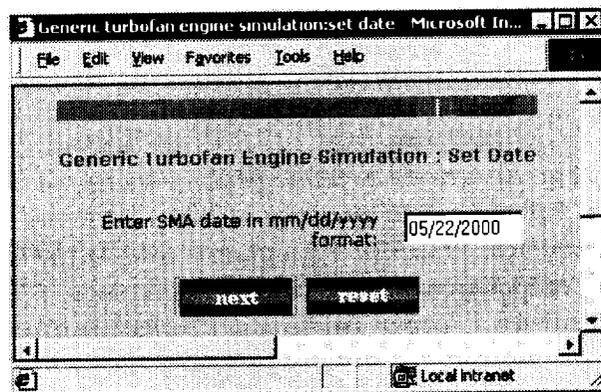


Figure 7: Select SMA date.

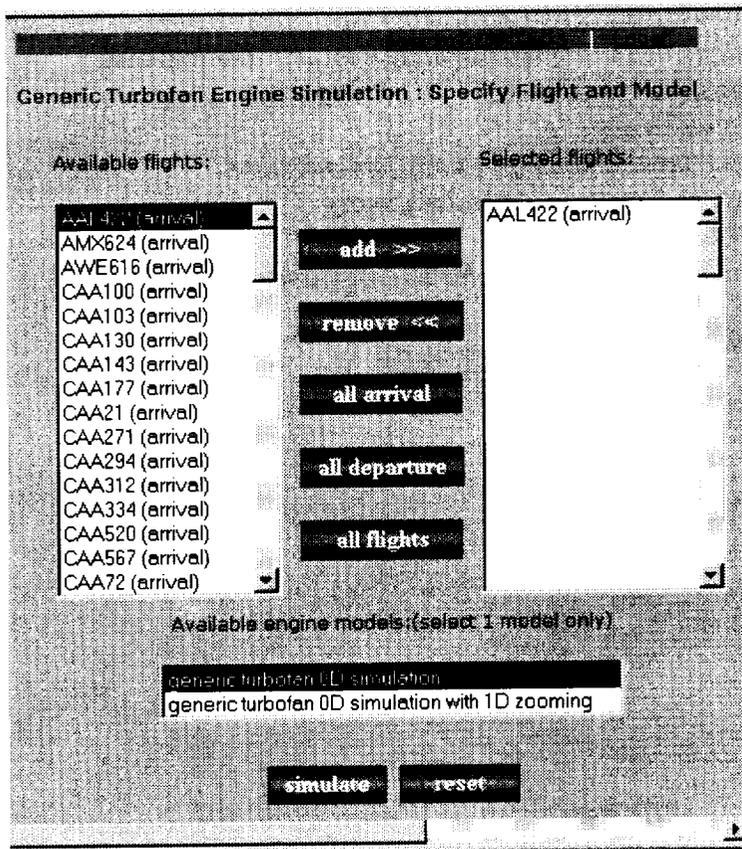


Figure 8: Select flights and engine model.

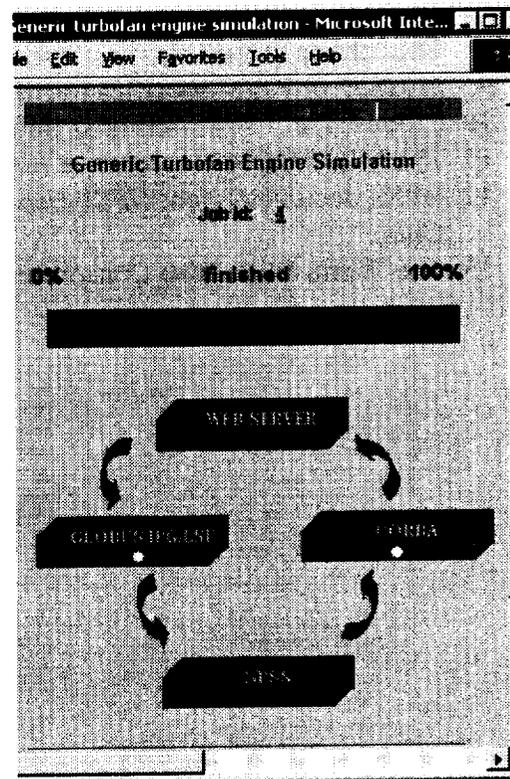


Figure 9: Applet showing job status.

Future Work

XML is becoming the de facto standard format for data exchange [13]. Both flight input data and engine simulation output will be XML format. Document type definitions (DTD) for the format will be defined. Use of XML will facilitate the data flowing in a uniform way but being processed in arbitrary ways.

Another development will be using fast-time data. Instead of storing static input data locally, a CORBA server running on a coupling machine will retrieve it from a remote SMA Data Server upon request.

References

1. J. Bardina, G. J. Follen, et al. *Integrated airplane health management system*. First JANNAF Modeling and Simulation Subcommittee Meeting, 11/00
2. G. Follen, C. Naiman. *Numerical Propulsion Simulation System (NPSS)/National Cycle Program (NCP)*. HPCC Computational Aerosciences Conference, 8/98
3. G. Follen, A. Evans, C. Naiman, I. Lopez. *Numerical Propulsion Simulation System*. AIAA 98-3113, 7/98
4. NASA's Numeric Propulsion System Simulation (NPSS) task at <http://hpcc.lerc.nasa.gov/hpcc2/npssintro.shtml>
5. IPG project at <http://www.ipg.nasa.gov>
6. The Globus project at <http://www.globus.org>
7. I. Foster, C. Kesselman. *Globus: A metacomputing infrastructure toolkit*. International Journal of Supercomputer Applications, 11:115-128, 1997
8. The Jakarta Project at <http://jakarta.apache.org>
9. JavaServer Pages specification at <http://java.sun.com/products/jsp/index.html>
10. CloudscapeTM from Informix Software at <http://www.cloudscape.org>
11. The MICO CORBA project at <http://www.mico.org>
12. NPSS documentation: *NPSS User Guide and Reference, NPSS Component Developer's Guide*
13. The World Wide Web Consortium (W3C) at <http://www.w3c.org>



REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE April 2001	3. REPORT TYPE AND DATES COVERED Technical Memorandum	
4. TITLE AND SUBTITLE Web-Based Distributed Simulation of Aeronautical Propulsion System			5. FUNDING NUMBERS WU-704-31-6300	
6. AUTHOR(S) Desheng Zhang, Gregory J. Follen, William R. Pavlik, Chan M. Kim, Xianyou Liu, Tammy M. Blaser, and Isaac Lopez				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration John H. Glenn Research Center at Lewis Field Cleveland, Ohio 44135-3191			8. PERFORMING ORGANIZATION REPORT NUMBER E-12733	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001			10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA TM-2001-210818	
11. SUPPLEMENTARY NOTES 10th International Symposium on High Performance Distributed Computing sponsored by the Institute of Electrical and Electronics Engineers, San Francisco, California, August 7-9, 2001. Desheng Zhang, William R. Pavlik, and Xianyou Liu, AYT Corporation, 2001 Aerospace Parkway, Brook Park, Ohio 44142 (work funded by NASA Contract NAS3-00180); Gregory J. Follen, Chan M. Kim, Tammy M. Blaser, and Isaac Lopez, NASA Glenn Research Center. Responsible person, Gregory J. Follen, organization code 2900, 216-433-5193.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category: 61 Available electronically at http://gltrs.grc.nasa.gov/GLTRS This publication is available from the NASA Center for AeroSpace Information, 301-621-0390.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) An application was developed to allow users to run and view the Numerical Propulsion System Simulation (NPSS) engine simulations from web browsers. Simulations were performed on multiple INFORMATION POWER GRID (IPG) test beds. The Common Object Request Broker Architecture (CORBA) was used for brokering data exchange among machines and IPG/Globus for job scheduling and remote process invocation. Web server scripting was performed by JavaServer Pages (JSP). This application has proven to be an effective and efficient way to couple heterogeneous distributed components.				
14. SUBJECT TERMS Computer techniques			15. NUMBER OF PAGES 14	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	



